Two-Way ANOVA Theoretical and Practical Calculations: In-Class Exercise

Key

Psychology 311
Spring, 2013

The e-book *[R] Companion for Experimental Design and Analysis for Psychology* by Williams, Krishnan, and Abdi contains the following example.

> Consider a replication of an experiment by Tulving & Pearlstone (1966), in which 60 subjects were asked to learn lists of 12, 24 or 48 words (factor A with 3 levels). These words can be put in pairs by categories (for example, apple and orange can be grouped as "fruits"). Subjects were asked to learn these words, and the category name was shown at the same time as the words were presented. Subjects were told that they did not have to learn the category names. After a very short time, subjects were asked to recall the words. At that time half of the subjects were given the list of the category names, and the other half had to recall the words without the list of categories (factor B with 2 levels). The dependent variable is the number of words recalled by each subject. Note that both factors are fixed. (p. 192)

The data are presented in the following table:

	Factor $\mathcal{A}$					
Factor $\mathcal{B}$	a_1: 12 words		a_2: 24 words		a_3: 48 words	
	11	07	13	15	17	16
	09	12	18	13	20	23
b_1	13	11	19	09	22	19
Free Recall	09	10	13	08	13	20
	08	10	08	14	21	19
	12	10	13	14	32	30
	12	12	21	13	31	33
b_2	07	10	20	14	27	25
Cued Recall	09	07	15	16	30	25
	09	12	17	07	29	28

1. Our first step is to get the data into R in a form suitable for analysis. Create a data file containing the 60 scores, with appropriate factor levels. There are many ways you could do this. One way is to first enter the dependent variable scores directly into R, then add the group labels as `factor` variables. We need to be sure to keep track of the order in which we entered the data. One way is to subdivide the data according to the

"slowest moving" factor in the data set. To add the factor variables, you should take a close look at the help file for the gl function.

Answer. Start by setting up the dependent variable.

```
> free_recall <- c(11,9,13,9,8,7,12,11,10,10,13,18,19,13,8,15,13,9,8,14,
+                  17,20,22,13,21,16,23,19,20,19)
> cued_recall <- c(12,12,7,9,9,10,12,10,7,12,13,21,20,15,17,14,13,14,16,7,
+                  32,31,27,30,29,30,33,25,25,28)
> score <- c(free_recall,cued_recall)
```

Next, use the gl function to establish the grouping variables.

```
> # We now prepare the labels for the 3 x 2 x 10 scores according to
> # the factor levels:
> # Factor A --- 12 words 24 words 48 words, 12 words 24 words
> # 48 words, ... etc.
> list_length <- gl(3,10,2*3*10, labels=c("12 Words","24 Words",
+                                         "48 Words"))
> # Factor B --- Free Recall Free Recall , Cued Recall Cued
> # Recall etc.
> recall_type <- gl(2,3*10,2*3*10, labels=c("Free Recall",
+                                           "Cued Recall"))
```

Notice how the function works. The first input parameter is the number of levels of the factor. The second input parameter is the number of times to repeat each level. The third parameter is the total length of the sequence.You can also enter labels for each level.

Let's look at the first call to the function. The first call says that there are 3 levels, and each should be repeated only once for a sequence. The third parameter is the total sequence length. The total sequence length is 2*3*10, or 60, so the sequence of 3 items (each repeated once) should be duplicated 20 times.

Next, we convert these lists into factor variables and create a data frame all in one command.

```
> mem.data <- data.frame(score, list_length,recall_type)
> mem.data

   score list_length recall_type
1     11    12 Words Free Recall
2      9    12 Words Free Recall
3     13    12 Words Free Recall
4      9    12 Words Free Recall
5      8    12 Words Free Recall
```

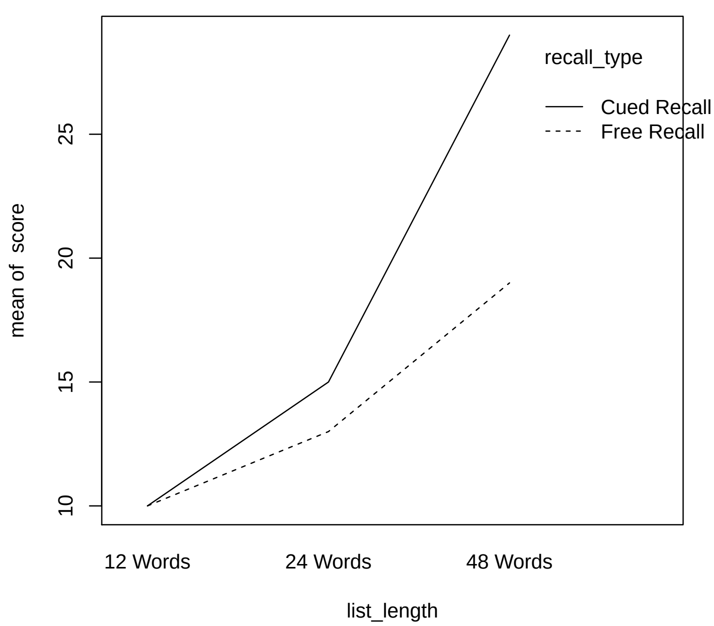
```
6      7    12 Words Free Recall
7     12    12 Words Free Recall
8     11    12 Words Free Recall
9     10    12 Words Free Recall
10    10    12 Words Free Recall
11    13    24 Words Free Recall
12    18    24 Words Free Recall
13    19    24 Words Free Recall
14    13    24 Words Free Recall
15     8    24 Words Free Recall
16    15    24 Words Free Recall
17    13    24 Words Free Recall
18     9    24 Words Free Recall
19     8    24 Words Free Recall
20    14    24 Words Free Recall
21    17    48 Words Free Recall
22    20    48 Words Free Recall
23    22    48 Words Free Recall
24    13    48 Words Free Recall
25    21    48 Words Free Recall
26    16    48 Words Free Recall
27    23    48 Words Free Recall
28    19    48 Words Free Recall
29    20    48 Words Free Recall
30    19    48 Words Free Recall
31    12    12 Words Cued Recall
32    12    12 Words Cued Recall
33     7    12 Words Cued Recall
34     9    12 Words Cued Recall
35     9    12 Words Cued Recall
36    10    12 Words Cued Recall
37    12    12 Words Cued Recall
38    10    12 Words Cued Recall
39     7    12 Words Cued Recall
40    12    12 Words Cued Recall
41    13    24 Words Cued Recall
42    21    24 Words Cued Recall
43    20    24 Words Cued Recall
44    15    24 Words Cued Recall
45    17    24 Words Cued Recall
46    14    24 Words Cued Recall
47    13    24 Words Cued Recall
48    14    24 Words Cued Recall
49    16    24 Words Cued Recall
50     7    24 Words Cued Recall
51    32    48 Words Cued Recall
```

```
52    31    48 Words Cued Recall
53    27    48 Words Cued Recall
54    30    48 Words Cued Recall
55    29    48 Words Cued Recall
56    30    48 Words Cued Recall
57    33    48 Words Cued Recall
58    25    48 Words Cued Recall
59    25    48 Words Cued Recall
60    28    48 Words Cued Recall
```

2. Let's look at the interaction plot. Imagine for a moment that the plot displayed population cell means. Which main effects, simple main effects, and interactions are *zero*?

   Answer. The means appear to be the same for both recall types at the 12 Word level of `list_length`. So if these were population means, we would say that there is no simple main effect of recall type at the 12 Word level of `list_length`. However, all other effects are non-zero.

   ```
   > interaction.plot(list_length,recall_type,response=score)
   ```



3. Next, perform the two-way ANOVA using the `aov` function.

Answer.

```
> aov1 <- aov(score~list_length*recall_type, data=mem.data)
> print(model.tables(aov1,"means"),digits=3)

Tables of means
Grand mean

16

 list_length
list_length
12 Words 24 Words 48 Words
      10       14       24

 recall_type
recall_type
Free Recall Cued Recall
         14          18

 list_length:recall_type
           recall_type
list_length Free Recall Cued Recall
   12 Words 10          10
   24 Words 13          15
   48 Words 19          29

> summary(aov1)

                        Df Sum Sq Mean Sq F value   Pr(>F)
list_length              2   2080    1040  115.56  < 2e-16 ***
recall_type              1    240     240   26.67 3.58e-06 ***
list_length:recall_type  2    280     140   15.56 4.62e-06 ***
Residuals               54    486       9
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4. Use the quick calculation method for main effects to calculate the F test for the main effect of `list_length`.

   Answer. The quick calculation method for an equal n 2-way ANOVA involves taking the row or column means, and treating them as if they were the means in a 1-Way ANOVA, with an "effective n" equal to the number of each observations in each row (or column).

   Consider, for example, the column effect in this case. Since there are two rows, and $n = 10$ per cell, each column mean was based on a total of 20 observations. So the effective n is 20 for the column effect. The 3 column means are 10,14,24.

```
> ss <- var(c(10,14,24))
```

They have a variance of 52. The residual mean square is 9. The F statistic is therefore

$$F = \frac{S^2_{\bar{X}}}{\hat{\sigma}^2/n_{\text{effective}}} = \frac{52}{9/20} = 1040/9 = 115.56 \tag{1}$$

5. Test the significance of the simple main effect of `list_length` under "Free Recall."

Answer. We begin by grabbing one row of the data.

```
> library(xtable)
> fit.sme <- aov(score~list_length,data=subset(mem.data,
+                  recall_type=="Free Recall"))
> sme.table <- xtable(summary(fit.sme))
> print(sme.table)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
list_length	2	420.00	210.00	23.43	0.0000
Residuals	27	242.00	8.96		

```
> str(sme.table)
```

```
Classes 'xtable' and 'data.frame':        2 obs. of  5 variables:
 $ Df     : num  2 27
 $ Sum Sq : num  420 242
 $ Mean Sq: num  210 8.96
 $ F value: num  23.4 NA
 $ Pr(>F) : num  1.26e-06 NA
 - attr(*, "align")= chr  "l" "r" "r" "r" ...
 - attr(*, "digits")= num  0 0 2 2 2 4
 - attr(*, "display")= chr  "s" "f" "f" "f" ...
```

We can re-do the test using a mean square residual from all the data (9.00, from the previous ANOVA table) if we want. One of the easiest ways to do this is to grab the results and simply insert them into the table. I'm showing here how do do this in LaTeX, but the same principles hold for ordinary R object. Above, I list the structure for the `xtable` object. Now I simply insert the numbers.

```
> pvalue <- 1-pf(210/9,2,54)
> sme.table2 <- sme.table
> sme.table2$Df[2] <- 54
> sme.table2$'Sum Sq'[2] <- 486
> sme.table2$'Mean Sq'[2] <- 9
> sme.table2$'F value'[1] <- 210/9
> sme.table2$'Pr(>F)'[1] <- pvalue
> print(sme.table2)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
list_length	2	420.00	210.00	23.33	0.0000
Residuals	54	486.00	9.00		

6. Construct a confidence interval for ω^2 for the main effect of `list_length`.

Answer. In the course lecture notes, we have

$$\omega^2 = \frac{\lambda}{\lambda + N_{tot}}$$

We can use this relationship to construct a confidence interval on ω^2 from a confidence interval on the noncentrality parameter λ.

```
> library(MBESS)
> out <- conf.limits.ncf(115.56,df.1=2,df.2=54)
> out

$Lower.Limit
[1] 136.2028

$Prob.Less.Lower
[1] 0.025

$Upper.Limit
[1] 346.6059

$Prob.Greater.Upper
[1] 0.025

> N.tot <- 60
> Lower.Limit <- out$Lower.Limit /(out$Lower.Limit + N.tot)
> Upper.Limit <- out$Upper.Limit /(out$Upper.Limit + N.tot)
> Lower.Limit

[1] 0.694194
```

```
> Upper.Limit
[1] 0.852437
```

7. We can, of course, "go the other way" to compute the power to detect an effect corresponding to a particular value of ω^2 by using the above equation to convert ω^2 to λ. In this case, suppose that $n = 10$ per cell in a 2×3 2-way ANOVA, and that $\omega^2 = .60$. What would be the power?

   Answer. With a little manipulation, we can express λ as a function of N_{tot} and ω^2. We get

   $$\lambda = N_{tot}\frac{\omega^2}{1-\omega^2}$$

   ```
   > print(lambda <- N.tot * (0.60/(1-0.60)))
   [1] 90
   ```

   Next, we compute the power directly, using the standard assumption that $\alpha = 0.05$.

   ```
   > 1 - pf(qf(0.95,2,54),2,54,90)
   [1] 1
   ```

   We can see that the power would be extremely high.

8. After observing a particular value of F, is it possible to construct a confidence interval on what the value of power was in the experiment just performed?

   Answer. Yes. Since F yields a confidence interval on λ, and λ corresponds, in a given analysis, to power in a 1-1 functional relationship, a confidence interval on λ leads directly to a confidence interval on power.